

A Method to Speed-Up Retinex-Type Algorithms

Technical Field

The technical field is color vision correction, and more particularly color vision correction using Retinex-type algorithms.

5 Background

A theory of human vision centered on the concept of a center/surround Retinex (retina and cortex) was introduced by Edwin Land in "An Alternative Technique for the Computation of the Designator in the Retinex Theory of Color Vision," Proceedings of the National Academy of Science, Volume 83, pp. 3078-3080, 1986. Land drew upon his
10 earlier Retinex concepts disclosed in "Color Vision and The Natural Image," Proceedings of the National Academy of Science, Volume 45, pp. 115-129, 1959, but harmonized these with certain findings of the neurophysiology of vision. All of the Retinex concepts were intended to be models for human color perception.

The application of Land's human vision theories to image processing has been
15 attempted in the prior art. For example, to mimic the dynamic range compression of human vision, a detector array with integrated processing in analog VLSI silicon chips used a logarithm transformation prior to the surround formation. See "Analog VLSI and Neural Systems," C. Mead, Addison-Wesley, Reading, Mass., 1989. In an attempt to improve color constancy, the implementation of a color Retinex in analog VLSI
20 technology is suggested by Moore et al., in "A Real-time Neural System for Color Constancy," IEEE Transactions on Neural Networks, Volume 2, pp. 237-247, March 1991. In Moore et al., the surround function was an exponential and final processing before display of the image required the use of a variable gain adjustment that set itself by finding the absolute maximum and minimum across all three color bands' signal values.

25 Central to these and other prior art Retinex methods is a Retinex-type algorithm. A perceived image S is a multiplication between the illumination L shed on visible surfaces and the respective reflectance R of the surfaces. Thus

$$(1) \quad S = R \cdot L.$$

An underlying assumption behind Retinex algorithms is that the illumination L is
30 an artifact. The illumination L is estimated and either removed completely: $S_1 = S/L = R$; or partially: $S_2 = S/f(L)$, where $f(L)$ is a function of the illumination. Estimating L from S is the main algorithmic and computational problem in Retinex algorithms.

Prior art Retinex-type algorithms are characterized by a two-module structure as shown in Figure 1. A local statistics module 10 computes a smooth version L (i.e., the

local statistics) of an input image S. Usually the smooth version L is either a linear or a non-linear low-pass filter of the input image S. A manipulation module 20 then manipulates pixels in image S according to correspondingly located values in the local statistics L. In the case of Retinex-type algorithms, the local statistics L is usually attributed to the illumination, and is a local average or local envelope (maximum) of S. Sometimes the local statistics L may be a robust local average or robust local envelope of S, whereby robust means that pixels participating in determining the local average or envelope are on the same side of perceptually significant image edges as the center pixel. In the discussion that follows, L will be referred to as illumination; however L should be understood to encompass its more general meaning.

In Figure 1, for convenience, the input image S is shown as the input to the Retinex algorithm. However, as is known to those of ordinary skill in the art, Retinex-type algorithms typically operate in the Log domain. As is also known to those of ordinary skill in the art, the illumination L is often referred to as an "envelope." The envelope can be smooth or piece-wise smooth.

Prior art Retinex algorithms also typically use linear space invariant low pass filters or partial differential equations for the illumination estimation module 10. Variants include slowly varying envelopes, i.e., local envelopes instead of local averages, and robust low passes resulting in piece-wise smooth averages or envelopes, which might change abruptly whenever the input changes abruptly.

In the illumination manipulation module 20 module, the illumination L might be subtracted in part, for example, subtract half of the illumination L from the input image S. Alternative manipulation methods may reduce more of the input image S values as corresponding illumination L values increase.

Prior art Retinex algorithms may be applied to monochrome or color images. In the color image case, the Retinex algorithm may be applied to all planes, or only to the illumination L (e.g., the Y) plane.

In some prior art Retinex algorithms both the illumination estimation and illumination manipulation modules 10 and 20 are performed in an iterative filtering and subtraction scheme. In other prior art Retinex algorithms, the modules 10 and 20 are interleaved in a scale-space.

In an improved prior art Retinex-type algorithm, the illumination L is obtained from a sub-sampled version of the input image S. Such an improved Retinex-type algorithm is shown in block diagram form in Figure 2. In Figure 2, the Retinex-type

algorithm includes linear illumination estimation module 30 and the illumination manipulation module 20. The image signal S is first input to a down sample module 32, where the image signal S is sub-sampled using techniques known to those of ordinary skill in the art to produce sub-sampled images S' . For example, the sub-sampling may involve averaging blocks of four pixels in the image S . The linear illumination estimation module 30 then generates an estimate of the illumination L' based on the sub-sampled image S' . The illumination L' is then provided to an up-sample module 34, and an estimation of the illumination L'' of the entire image S is produced, using interpolation and similar techniques known to those of ordinary skill in the art.

Sub-sampling and subsequent interpolation are intended to speed up the computationally intensive Retinex process. However, in the Retinex-type algorithm illustrated in Figure 2, the interpolation is performed on a set of smooth, low resolution intermediate images (i.e., the images S'), using the high resolution input image S to select corresponding output pixels from the resulting set of high resolution images L'' . This interpolation scheme avoids the need to interpolate a low resolution piece-wise smooth illumination image. Thus, the Retinex-type algorithm shown in Figure 2 can only be used where the algorithm's computationally intensive operation produces a smooth function (either an average or an envelope). An example of such a Retinex-type algorithm is described in F. Durand, and J. Dorsey, "Fast Bilateral Filtering for the Display of High Dynamic Range Images", preprint in <http://graphics.lcs.mit.edu/~fredo/DurandBilateral.pdf>. Thus, the Retinex-type algorithm shown in Figure 2 cannot be used for robust Retinex-type algorithms, other than for those Retinex-type algorithms employing bilateral filtering. Furthermore, the computationally intensive operation of the Retinex-type algorithm is repeated several times, once for each of the intermediate images S' .

25 Summary

What is disclosed is an apparatus for speeding up Retinex-type processing of an input image. The apparatus includes a down sample module having a sub-sampling algorithm, where sub-sampled images of the input image are produced, and a non-linear illumination estimation module that receives the sub-sampled images and produces corresponding interim illumination estimations. Finally, the apparatus includes an up sample module including one or more up-sampling algorithms. The interim illumination estimations are interpolated to produce an illumination estimation, and the illumination estimation is usable to perform a Retinex correction to the input image.

Also disclosed is a method for speeding up Retinex-type processing of a high resolution input image. The method includes the steps of sub-sampling the high resolution input image to produce one or more low resolution input images, estimating an illumination of the low resolution images, where an interim illumination estimation is generated for each low resolution input image, and up-sampling the interim illumination estimation, where an illumination estimation suitable for Retinex correction is generated.

Description of the Drawings

The detailed description will refer to the following drawings in which like numbers refer to like items, and in which:

10 Figures 1 and 2 are block diagrams of prior art Retinex-type algorithms;

 Figure 3 is a block diagram of an embodiment of a Retinex-type algorithm that is an improvement over the prior art;

 Figures 4a and 4b are block diagrams of two possible up-sampling algorithms used with the Retinex algorithm of Figure 3;

15 Figures 5a – 5c are schematic illustrations of signals through the two up-sampling algorithms of Figures 4a and 4b;

 Figures 6a – 6f present an example of a sub-sampled non-linear Retinex algorithm;

20 Figures 7a and 7b are block diagrams of up-sampling combinations to balance blurring artifacts and over-sharpening artifacts; and

 Figures 8a – 8c present results of combining up-sampling algorithms compared to the results of using the up-sampling algorithms individually.

Detailed Description

25 The description that follows will use the term “linear” according to one of two in two different meanings:

1. Linear algorithms (e.g., a linear convolution) as opposed to non-linear algorithms.
2. Linear computational complexity as opposed to, for example, square or logarithmic complexity. Specifically, the description will refer to linear or sub-linear complexity with respect to the input complexity, namely, is the number of computationally intensive operations proportional to the number of input pixels, or is it proportional to that number divided by some sampling factor.

30 A major problem with using robust Retinex algorithms on sub-sampled data is that the piece-wise smooth illumination estimation results of robust Retinex algorithms do not interpolate well, and artifacts are likely in the output illumination. To overcome this

problem, an improved robust Retinex algorithm disclosed herein avoids artifacts when a robust Retinex illumination estimation is interpolated, thereby enabling a significant processing speed-up over prior art Retinex algorithms.

In both robust and non-robust illumination estimation modules sub-sampling
5 presents a question of whether the illumination estimation L' of the sub-sample S' is a local maximum or a local average. In robust illumination estimation modules, artifacts are introduced during up-sampling, or interpolation. However, when the sampling factor is small relative to the size and viewing distance of the output image, the artifacts sharpen the output image compared to non-sub-sampled equivalents. Thus, up to a certain extent,
10 these artifacts can be considered as an advantage. In robust illumination estimation modules, when sub-sampling is relatively strong, over-sharp and halo artifacts created during the required up sampling may degrade the quality of the output image. In these cases an alternative, non-standard interpolation approach, which will be described in detail later, may be used in the up-sampling stage. However, this alternative up-sampling
15 creates blurring artifacts. Finally, the up-sampling algorithms may be combined in one of at least the following ways to balance the sharpening of one interpolation algorithm against the blurring of another interpolation algorithm:

Average the two up-sampling algorithms according to the required total sharpening.

20 Averaging-rate changes throughout the image, according to local properties of the image S .

Cascade the two up-sampling algorithms. The resulting relative interpolation rate determines the total sharpening.

25 Cascaded rate changes throughout the image, according to the local properties of the image.

Figure 3 is a block diagram of an embodiment of an improved Retinex algorithm
100 that speeds up Retinex processing. The Retinex algorithm 100 processes input signal S and produces output signal R . The Retinex algorithm 100 includes down-sampling module 110, which receives the input signal S and performs down-sampling according to
30 a down-sampling algorithm (not shown in Figure 3) to produce one or more sub-sampled images S' . The sub-sampled images S' are then provided to non-linear illumination estimation module 120, which produces an interim illumination estimation L' for each of the sub-sampled images S' . The interim illumination estimations L' are provided to up-sampling module 140, which uses an up-sampling algorithm (not shown in Figure 3) to

produce a full size illumination estimation L'' . The illumination estimation L'' is then provided to illumination manipulation module 180, which also receives the input signal S . The illumination manipulation module 180 processes the received illumination estimation L'' and the input signal to produce output signal R .

5 The down-sampling module 110 filters and sub-samples the input image S . In the case where the illumination L is an envelope rather than a local average, the sub-sampling performed by the down-sampling module 110 may be a sampling of a local maximum of the input image S . Alternatively, for large sampling factors, a combination of sub-sampling algorithms may be used. For example, sub-sampling by the down-sampling
10 module 110 may include sampling by a factor of s_1 using averaging (local-maximum), and by a factor of s_2 using local-maximum (averaging), for an overall sampling factor of $S = s_1 * s_2$.

 When the non-linear illumination estimation L is a non-robust envelope, local maximum sampling may be preferable. Averaging reduces the contrast of small bright
15 structures (in this context, small is relative to the sampling factor). In some cases, this type of artifact may be preferred in order to reduce potential specularities in the output image R . In this case, averaging should not exceed the expected size of the specularities. When the sampling factor is larger, the remaining portions of the input image S should be down sampled using local maxima.

20 In case the non-linear illumination estimation L is a robust envelope, using local maximum sampling might increase halo artifacts, as discussed above. The combination of this constraint on the local-maximum-based sampling factor, and the constraint on the averaging-based sampling factor may limit the sampling factor of non-robust envelope modules.

25 The up-sampling module 140 uses an up-sampling algorithm as known in the art of image and signal processing. In some up-sampling algorithms and in the appropriate up-scaling rates, new samples are interpolated between low-resolution input samples. Standard up-sampling algorithms include nearest neighbor, bi-linear, and bi-cubic interpolations. However, in the Retinex algorithm 100, details in the high-resolution
30 output image R cannot mismatch details in the original input image S (inaccurately placed details will result in artifacts). The solution to this problem is to design the up-sampling algorithm so that the interpolation does not add details, or else use the input image S as a guide to detail placement.

Figure 4 details the two design options for the algorithm 150. In Figure 4a, illumination interpolation algorithm 150' includes interpolation routine 152 and local maximum routine 154. The interpolation routine 152 receives a low-resolution illumination estimation and a sampling rate, to produce high-resolution illumination values. The interpolation routine 152 may be, for example, bilinear interpolation. Other interpolation methods may be used with the interpolation routine 152 so long as those methods do not estimate or create new details. When the estimated illumination L is an envelope rather than an average, and in rare cases in which the interpolated illumination will be below the high resolution input image, the envelope constraint is enforced explicitly using the maximum routine 154.

Figure 4b is a block diagram of the difference interpolation-algorithm 150''. The algorithm 150'' includes the interpolation routine 152, which performs the same functions as described above with respect to the algorithm 150'. However, the algorithm 150'' interpolates a high-resolution difference (between the illumination L and the input image S) and low-resolution values using the same interpolation methods as described above with respect to Figure 4a. The algorithm 150'' includes adder 153, which combines the low-resolution input (i.e., the sub-samples of the input image) and the low resolution interim illumination estimation L. The combined value is then provided to the interpolation routine 152. The interpolated difference output is then combined in adder 155 with the high resolution image input.

Figure 5 is a schematic illustration of signals through the two interpolation algorithms 150' and 150''. In Figures 5a, 5b and 5c, the high-resolution input image S is shown as curve 201, and the corresponding low resolution envelope L is shown as curve 203. Low-resolution samples are shown as dots 205 in Figure 5b, and are interpolated to form the piece-wise constant curve 203. Piece-wise linear interpolation, which might result from the algorithm 150' described in Figure 4a is shown in Figure 5a as curve 207. Overlapping curve 207 is a final output image R showing the results of clipping, and depicted as curve 209. For clarity, the curves 207 and 209 have been slightly displaced, in Figure 5a.

Figure 5b shows the signals for difference interpolation algorithm 150''. Besides the high resolution input image curve 201 and the low resolution envelope curve 203, a difference interpolation is shown as curve 211.

Comparing the signals in Figures 5a and 5b shows that the linear interpolation curve 207 of Figure 5a has no new details, whereas the difference interpolation curve 211

of Figure 5b has many of the input image details (i.e., the curve 211 is in good alignment with the input image curve 201).

Returning to Figure 3, the illumination manipulation module 180 executes routines that essentially produce a difference between the illumination estimation L and the input image S. As a result, the output R of the illumination manipulation module 180 will have properties that are inverted from those of the illumination image L. This condition is shown schematically in Figure 5c, where the curve 213 illustrates that the output of an illumination interpolation algorithm 150' is considerably sharper than an output of the difference interpolation algorithm 150'', shown as curve 215.

An example of a sub-sampled non-linear Retinex algorithm is shown in Figure 6, which shows various parts of a patio, where the interior part includes a back wall, an altar, and a fence, among other features. Figure 6a presents an input image. Figure 6b presents a zoomed-in part of the input image of Figure 6a. Figure 6c presents the 1:5 illumination-interpolation of the estimated illumination of the image sub-sampled (1:5) from the zoomed-in part shown in Figure 6b. Figure 6d presents a Retinex correction corresponding to Figure 6c. Figure 6e presents the 1:5 difference interpolation of the estimated illumination of the image sub-sampled (1:5) from the zoomed-in part of the input image of Figure 6b. Figure 6f presents a Retinex correction corresponding to Figure 6e. Note that the sharper Retinex output shown in Figure 6d corresponds to the blurred interpolated illumination of Figure 6c, and the less sharp Retinex output shown in Figure 6f corresponds to the sharper difference interpolation of Figure 6e.

Choice of the interpolation algorithm depends on imaging intent and required interpolation rate. While usually sharp images are desirable, in some applications, over sharpening is a problem. Furthermore, for large sampling rates, artifacts become halos rather than sharpening artifacts. In all these cases, the blurring artifacts of difference interpolation can be balanced against the artifacts of illumination interpolation. Alternatives include: 1) averaging the illumination images from illumination and difference interpolations; and 2) cascading the illumination and difference interpolations, such that the total interpolation rate is the required interpolation.

Figures 7a and 7b are block diagrams of the two alternative algorithms, average algorithm 220 and cascade algorithm 230, respectively, for balancing artifacts. For algorithm 230 to perform the required up scaling, $R_0 = R_I * R_D$, where R_0 is an overall sampling rate, R_I is an interpolation sampling rate, and R_D is a difference sampling rate.

In Figure 7a, the average algorithm 220 uses average routine 157 to obtain a weighted average of the illumination and difference interpolation algorithms 150' and 150''. The average routine 157 applies weight W_D to the difference interpolation result and weight W_I to the illumination interpolation result. Since $W_I + W_D = 1$, the weights W_i play a similar role to rate ratios R_i/R_0 in algorithm 230.

In Figure 7b, cascade algorithm 230 first applies the difference interpolation algorithm 150'' to the low-resolution illumination followed by the illumination interpolation algorithm 150'.

Figure 8a shows the result of application of the illumination interpolation algorithm 150' and Figure 8b shows the result of application of the difference interpolation algorithm 150''. Figure 8c shows the result of applying the average algorithm 220, with $W_I = W_D = 0.5$.

Different ways to combine the two interpolation algorithms 150' and 150'' may be motivated by the fact that different artifacts are significant in different locations in an image. For example, halos are more visible near strong edges for which one side is in the mid-tones (high and low tones have less distinctive halos). This is evident in Figures 6c and 6d, where the halos on the dark altar-piece are less noticeable than those on the back wall. Also, at texture regions halos are much less visible, as, for example, in the case of the textured dark fence against the wall versus the edge between the altar-piece and the wall. In such cases it might be preferable to modify the rates R_i or the weights W_i locally, according to image properties.